自動残差修正機能付きGBiCGSTAB(s,L)法 (科学技術計算アルゴリズムの数理的基盤と展開)

AUTHOR(S):

塚田, 健; 深堀, 康紀; 谷尾, 真明; 杉原, 正顯

CITATION:

塚田, 健 ...[et al]. 自動残差修正機能付きGBiCGSTAB(s,L)法 (科学技術計算アルゴリズムの数理的基盤と展開). 数理解析研究所講究録 2011, 1733: 149-159

ISSUE DATE:

2011-03

URL:

http://hdl.handle.net/2433/170735

数理解析研究所講究録
第1733巻 2011年 149-159

自動残差修正機能付き GBiCGSTAB(s,L) 法

GBiCGSTAB(s,L) with Auto-Correction of Residuals

新日鉄ソリューションズ　塚田健 (Takeshi TSUKADA)
NS Solutions Corporation
東京大学大学院 情報理工学系研究科　深堀康紀 (Kouki FUKAHORI)
Graduate School of Information Science and Technology
The University of Tokyo
NEC　谷尾真明 (Masaaki TANIO)
NEC Corporation
東京大学大学院 情報理工学系研究科　杉原正顯 (Masaaki SUGIHARA)[*1]
Graduate School of Information Science and Technology
The University of Tokyo

1 はじめに

線形方程式

$$Ax = b$$

を考える．ここで A は与えられた $N \times N$ 非対称実行列，b は与えられた N 次元ベクトルである．2007年，Sonneveld と van Gijzen [6, 7] は，この線形方程式に対する新しい数値解法，IDR(s) 法を提案した．この解法は，高々 $N + N/s$ 回の行列ベクトル積によって真の解を与えるという理論上の特長を持ち，多くの数値実験において，BiCG 法系の解法と同等もしくはそれ以上の収束性能をもつことが知られていた．しかしながら，係数行列が歪対称に近い場合には，BiCGSTAB(L) 法 ($L > 1$) に大きく劣ることも知られていた．この弱点は，IDR(s) 法に含まれる安定化多項式の次数が1であり，一方，BiCGSTAB(L) 法の場合は $L > 1$ であることによる．この弱点を克服すべく，2009年に，IDR(s) 法に高次の安定化多項式を組み込んだ解法，GBiCGSTAB(s, L) 法 [8] および IDR(s)stab(L) 法 [5] が提案された．GBiCGSTAB(s, L) 法と IDR(s)stab(L) 法は数学的には同値で，丸め誤差がない状況では同じ解を生成するが，アルゴリズムとしては異なるものである．ここでは，以下，GBiCGSTAB(s, L) 法について考える．

IDR(s) 法については，大きな s に対して偽収束 (アルゴリズムが生成する残差は小さいにも関わらず，真の残差は小さくない状況) がしばしば起こることが報告されていた．

[*1] E-mail: m_sugihara@mist.i.u-tokyo.ac.jp

GBiCGSTAB(s, L) 法もこの性質を引き継いでいる．IDR(s) 法については，櫻井等 [4] が残差を自動修正する，より正確には，偽収束を見つけ，Δr_k および $A\Delta x_k$ が満たすべき関係式:

$$\Delta r_k = -A\Delta x_k \tag{1}$$

ができるだけ成り立つように残差の計算を行う，計算量の少ない方法を開発しており，この方法を備えた IDR(s) 法を，彼等は AC-IDR(s) 法 (Auto-Corrected IDR(s)) とよんだ．

本稿では，同様の方法を GBiCGSTAB(s, L) 法に対して開発する．ここで開発した方法を備えた GBiCGSTAB(s,L) 法を AC-GBiCGSTAB(s,L) 法（GBiCGSTAB(s,L) with Auto-Correction of Residuals）と呼ぶことにする．また，比較のために，関係式 (1) を用いて残差を計算した GBiCGSTAB(s, L) 法も考え，この方法を DC-GBiCGSTAB(s, L) 法 (GBiCGSTAB(s, L) with Direct-Computaion of Residuals) と呼ぶことにする．オリジナルの GBiCGSTAB(s,L) 法，AC-GBiCGSTAB(s,L) 法，DC-GBiCGSTAB(s, L) 法を数値実験を通して比較する．

2 GBiCGSTAB(s, L) 法

GBiCGSTAB(s, L) 法のアルゴリズムは Algorithm 1 で与えられる．ここで

$$r_{k,p}^{(j)} = A^p r_k^{(j)},\ U_{k,p}^{(j)} = A^p U_k^{(j)}$$

と略記している．また，アルゴリズムは大きく 2 つの部分 GBiCG-PART と MR-PART からなる．

図 1 に GBiCGSTAB(8,8) を MatrixMarket にある問題 wang4 に適用した結果を示す (実線がアルゴリズムで生成される残差のノルム，破線が真の残差のノルムである)．偽収束が起こっていることが分かる．

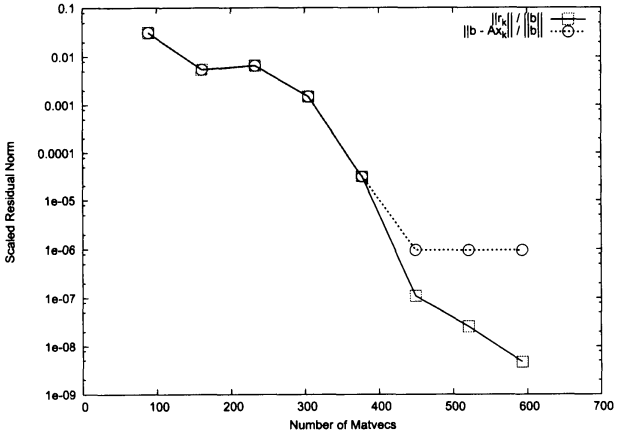


図 1 GBiCGSTAB(8,8) 法を MatrixMarket にある問題 wang4 に適用したときの残差履歴

Algorithm 1 GBiCGSTAB(s, L) 法

$x_0 \in \mathbb{R}^N$: given, $r_{0,0} := b - Ax_0$; $\tilde{R}_0 \in \mathbb{R}^{N \times s}$: given.
$k := 0$
Set $U_{0,0}^{(1)} := [r_0, Ar_0, \cdots, A^{s-1}r_0]$; Compute $U_{0,1}^{(1)}$;
$M_0 := \tilde{R}_0^T U_{0,1}^{(1)}$; $m_0 := \tilde{R}_0^T r_0$;
Solve $M\vec{\alpha}_0^{(1)} = m$ for $\vec{\alpha}_0^{(1)}$;
$r_{0,0}^{(1)} := r_{0,0} - U_{0,1}^{(1)}\vec{\alpha}_0^{(1)}$; $x_0^{(1)} := x_0 + U_{0,0}^{(1)}\vec{\alpha}_0^{(1)}$
while $\|r_{k,0}^{(0)}\| \geq \epsilon\|b\|$ **do**
  /*GBiCG-PART*/
  for $j = 1$ to L do **do**
    if $(k = 0) \cap (j = 1)$ **then**
      Go to { $j = 2$ }
    end if
    for $i = 1$ to s **do**
      if $i = 1$ **then**
        Solve $M_k^{(j-1)}\vec{\beta} = m_k^{(j-1)}$ for $\vec{\beta}$;
        $U_{k,p}^{(j)}e_1 := r_{k,p}^{(j-1)} - U_{k,p}^{(j-1)}\vec{\beta} \quad (p = 0, 1, \cdots, j-1)$
      else
        Solve $\left[m_k^{(j-1)}, M_k^{(j)}[1 : i-2], M_k^{(j-1)}[i : s]\right]\vec{\beta} = M_k^{(j)}e_{i-1}$ for $\vec{\beta}$;
        $U_{k,p}^{(j)}e_i := U_{k,p+1}^{(j)}e_{i-1} - \left[r_{k,p}, U_{k,p+1}^{(j)}[1 : i-2], U_{k,p}^{(j-1)}[i : s]\right]\vec{\beta} \quad (p = 0, 1, \cdots, j-1)$
      end if
      Compute $U_{k,j}^{(j)}e_i = A \times U_{k,j-1}^{(j)}e_i$;
      $M_k^{(j)}e_i := \tilde{R}_0^\top U_{k,j}^{(j)}e_i$
    end for
    Solve $M_k^{(j)}\vec{\alpha}_k^{(j)} = m_k^{(j-1)}$ for $\vec{\alpha}_k^{(j)}$;
    $x_k^{(j)} := x_k^{(j-1)} + U_{k,0}^{(j)}\vec{\alpha}_k^{(j)}$;
    $r_{k,p}^{(j)} := r_{k,p}^{(j-1)} - U_{k,p+1}^{(j)}\vec{\alpha}_k^{(j)} \quad (p = 0, 1, \cdots, j-1)$;
    Compute $r_{k,j}^{(j)} = A \times r_{k,j-1}^{(j)}$
  end for
  /*MR-PART*/
  $\vec{\gamma}_{k+1} := \mathrm{argmin}_{\vec{\gamma}} \left\| r_{k,0}^{(L)} - \left[r_{k,1}^{(L)}, \cdots, r_{k,L}^{(L)}\right]\vec{\gamma} \right\|$;
  $r_{k+1,0}^{(0)} := r_{k,0}^{(L)} - \left[r_{k,1}^{(L)}, \cdots, r_{k,L}^{(L)}\right]\vec{\gamma}_{k+1}$;
  $x_{k+1}^{(0)} := x_k^{(L)} + \left[r_{k,0}^{(L)}, \cdots, r_{k,L-1}^{(L)}\right]\vec{\gamma}_{k+1}$;
  $U_{k+1,0}^{(0)} := U_{k,0}^{(L)} - \left[U_{k,1}^{(L)}, \cdots, U_{k,L}^{(L)}\right]\vec{\gamma}_{k+1}$;
  $M_{k+1}^{(0)} := -\gamma_{k+1,L}M_k^{(L)}$; $m_{k+1}^{(0)} := \tilde{R}_0^T r_{k+1}$;
  $k = k + 1$
end while

3 AC-GBiCGSTAB(s, L) 法

3.1 AC-IDR(s) 法

はじめにでも述べたが，櫻井等は偽収束を見つけ出し，残差を修正する機能をもつIDR(s)法，AC-IDR(s) 法を開発した.

彼等は，まず，Δr_k と $A\Delta x_k$ から計算される量 $\frac{\|\Delta r_k + A\Delta x_k\|}{\|b\|}$ （櫻井等は inconsistency と読んでいる）が偽収束の指標となることに注目した．しかし，この量の計算は重いので，inconsistency と相関が高く，かつ，計算量の少ない量を数値的に探し，それを偽収束の指標とした．ここでは，指標の具体形は省くが，この量を I_k と書くとき，自動残差修正機構は以下のようになる．ここで θ は残差を直接計算するかどうかの判定規準である.

Algorithm 2 自動残差修正機構

Compute Δx_k;
Compute I_k(偽収束の指標);
if ($I_k < \theta$) **then**
  Compute Δr_k normally;
else
  Compute Δr_k by using $\Delta r_k := -A\Delta x_k$;
end if
$x_{k+1} := x_k + \Delta x_k$;
$r_{k+1} := r_k + \Delta r_k$;

櫻井等はこの機構を取り入れた AC-IDR(s) 法が偽収束を克服でき，この機構が入ったことによる計算量増加は 3〜7 % であると報告している.

3.2 AC-GBiCGSTAB(s, L) 法

ここでは，AC-GBiCGSTAB(s, L) 法のアルゴリズムを与える.

GBiCGSTAB(s, L) 法においては，残差 Δr_k および近似解 Δx_k は次のように計算される：

$$\Delta x_k (= x_{k+1}^{(0)} - x_k^{(0)}) = \sum_{j=1}^{L} U_{k,0}^{(j)} \vec{\alpha}_k^{(j)} + \left[r_{k,0}^{(L)}, r_{k,1}^{(L)}, \cdots, r_{k,L-1}^{(L)} \right] \vec{\gamma}_{k+1},$$

$$\Delta r_k (= r_{k+1,0}^{(0)} - r_{k,0}^{(0)}) = -\sum_{j=1}^{L} U_{k,1}^{(j)} \vec{\alpha}_k^{(j)} - \left[r_{k,1}^{(L)}, r_{k,2}^{(L)}, \cdots, r_{k,L}^{(L)} \right] \vec{\gamma}_{k+1},$$

$$x_{k+1}^{(0)} = x_k^{(0)} + \Delta x_k,$$

$$r_{k+1,0}^{(0)} = r_{k,0}^{(0)} + \Delta r_k \ .$$

GBiCGSTAB(s, L) 法においても，IDR(s) の場合と同様に，Δr_k と $A\Delta x_k$ の inconsistency $\frac{\|\Delta r_k + A\Delta x_k\|}{\|b\|}$ が偽収束のよい指標となるが，上記の Δr_k および近似解 Δx_k の計算式から計算される量

$$I_k := \frac{\|r_{k,0}^{(0)}\|}{\|b\|} \times \max_{1\le j\le L} \left(\mathrm{Range}(\vec{\alpha}_k^{(j)})\right) \times \mathrm{Range}\,(\vec{\gamma}_{k+1})$$

(ただし，U の列ベクトルは正規化されているとする) が inconsistency と相関を持つことが数値的に確認される．ここで $\mathrm{Range}(\vec{c}) := \dfrac{\max\limits_{1\le i\le s}|c(i)|}{\min\limits_{1\le j\le s}|c(j)|}$ である．図 2 に I_k と inconsistency の実際の値を示す．比較的高い相関があることが見て取れる．

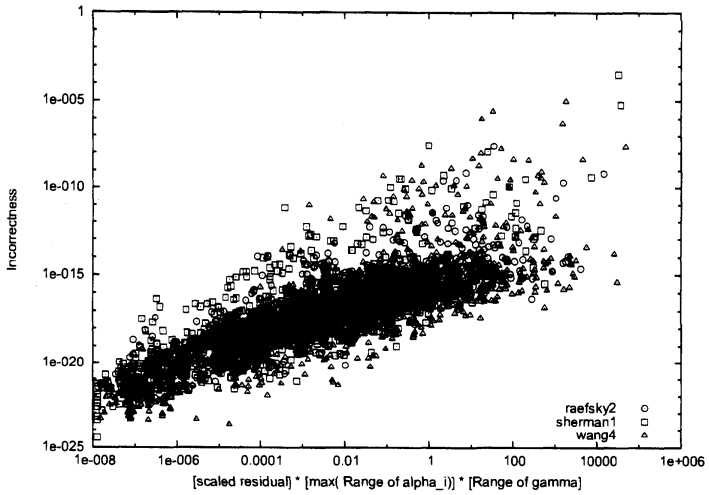


図 2 I_k と inconsistency の相関

GBiCGSTAB(s, L) 法に I_k を用いた自動残差修正機構を付加したアルゴリズム，AC-GBiCGSTAB(s, L) を Algorithm 3 に示す．

Algorithm 3 AC-GBiCGSTAB(s, L)

initial setting
while $\|r_{k,0}^{(0)}\| \geq \epsilon\|b\|$ **do**
/*GBiCG-PART*/
for $j = 1$ to L do **do**
if $(k = 0) \cap (j = 1)$ **then**
Go to { $j = 2$ }
end if
Update $U_{k,p}^{(j)}$ $(p = 0, 1, \cdots, j)$;
Compute $M_k^{(j)} = \tilde{R}_0^{\top} U_{k,j}^{(j)}$;
Solve $M_k^{(j)} \vec{\alpha}_k^{(j)} = m_k^{(j-1)}$ for $\vec{\alpha}_k^{(j)}$;
$r_{k,p}^{(j)} := r_{k,p}^{(j-1)} - U_{k,p+1}^{(j)} \vec{\alpha}_k^{(j)}$ $(p = 0, 1, \cdots, j-1)$;
Compute $r_{k,j}^{(j)} = A \times r_{k,j-1}^{(j)}$
end for
/*MR-PART*/
$\vec{\gamma}_{k+1} := \mathrm{argmin}_{\vec{\gamma}} \left\| r_{k,0}^{(L)} - \left[r_{k,1}^{(L)}, \cdots, r_{k,L}^{(L)} \right] \vec{\gamma} \right\|$;
$\Delta x_k := \sum_{j=1}^{L} U_{k,0}^{(j)} \vec{\alpha}_k^{(j)} + \left[r_{k,0}^{(L)}, r_{k,1}^{(L)}, \cdots, r_{k,L-1}^{(L)} \right] \vec{\gamma}_{k+1}$;
$I_k = \dfrac{\|r_{k,0}^{(0)}\|}{\|b\|} \times \max_j \left(\mathrm{Range}(\vec{\alpha}_k^{(j)}) \right) \times \mathrm{Range}\,(\vec{\gamma}_{k+1})$;
if ($I_k < \theta$) **then**
$\Delta r_k := -\sum_{j=1}^{L} U_{k,1}^{(j)} \vec{\alpha}_k^{(j)} - \left[r_{k,1}^{(L)}, r_{k,2}^{(L)}, \cdots, r_{k,L}^{(L)} \right] \vec{\gamma}_{k+1}$;
else
$\Delta r_k := -A \Delta x_k$; (direct-computation)
end if
$x_{k+1}^{(0)} := x_k^{(0)} + \Delta x_k$;
$r_{k+1,0}^{(0)} := r_{k,0}^{(0)} + \Delta r_k$;
end while

4 数値実験

ここでは，比較のために，関係式 (1) を用いて残差を計算した GBiCGSTAB(s, L) 法（この方法を DC-GBiCGSTAB(s, L) 法 (GBiCGSTAB(s, L) with Direct-Computaion of Residuals) と呼ぶことにする）も考え，オリジナルの GBiCGSTAB(s,L) 法，AC-GBiCGSTAB(s,L) 法，DC-GBiCGSTAB(s, L) 法を数値実験を通して比較する．

計算は Xeon E5450 processor (3.0GHz) 上で行い，本稿に書いた略式コードを Fortran90（コンパイラは Intel コンパイラ 10.1）で実装した．また，初期値等は以下のように設定した．

- 初期値： $x_0 = \mathbf{0}$.
- $s, L = 1, 2, 4, 8$.
- 終了条件： 相対残差ノルム 10^{-8} 以下，または，行列ベクトル積の数が $10N$（N は行列のサイズ）に達したとき.
- 偽収束の判定のためのパラメータ： $\theta = 0.1$.

実験に用いた行列は The University of Florida sparse matrix collection [1] および Matrix-Market [2] から，前処理無しで解が得られた行列 30 個を用いた (表 2 を参照). なお，方程式の右辺項 b は，解 x の成分がすべて 1 になるように選んだ.

表 1 に，行列 (a)wang4，(b) sme3Da の場合に，3 つの方法の行列ベクトル積の数，CPU time，反復が終了したときの真の相対残差を示す．真の相対残差 $\geq 10^{-8}$ のとき，すなわち偽収束のとき，真の相対残差をボールドで示した．この表より，GBiCGSTAB(s, L) 法については，$L = 8$ において偽収束が起きていること，一方 AC-GBiCGSTA(s, L) 法，DC-GBiCGSTAB(s, L) 法についてはほぼ偽収束が起きていないことが分かる．計算時間については，AC-GBiCGSTA(s, L) 法，DC-GBiCGSTAB(s, L) 法が GBiCGSTAB(s, L) 法より大きくなる場合が多いことが見て取れる.

偽収束のときの収束の様子を見るために，図 2，3 に，行列 wang4，sme3Da に GBiCGSTAB(8, 8) 法，AC-GBiCGSTAB(8, 8) 法，DC-GBiCGSTAB(8, 8) 法を適用したときの残差履歴を示す．図において，相対残差ノルムを実線で，真の相対残差ノルムを破線で表している．偽収束が比較的早い段階で始まっていることが分かる．また，AC-GBiCGSTAB 法と DC-GBiCGSTAB 法の収束履歴は，似ているものの微妙に異なることが分かる.

最後に，AC-GBiCGSTAB 法，DC-GBiCGSTAB 法の全般的有効性を見るために，表 2 に，30 個の行列に対して，GBiCGSTAB 法，AC-GBiCGSTAB 法，DC-GBiCGSTAB 法を適用したときの CPU time と真の解に収束した割合を示す．CPU time は GBiCGSTAB 法の CPU time を 1 にスケールして示してある．表 2 から， AC-GBiCGSTAB(s,L) 法，DC-GBiCGSTAB(s, L) 法において，GBiCGSTAB(s, L) 法のもつ数値的不安定性 (偽収束が起こること) が克服でき，計算時間も 10% 程度の増加に抑えることが出来ていることが分かる.

5 終りに

本稿では，GBiCGSTAB(s, L) 法のもつ数値的不安定性（偽収束が起こること）を克服するために自動残差修正機能付き GBiCGSTAB(s, L) 法（略して AC-GBiCGSTAB(s,L) 法）を開発し，数値実験を通してその有効性を調べた．その結果，数値的不安定性（偽収束が起こること）の克服に成功し，計算時間も 10% 程度の増加に抑えることが出来ることが分かった．しかし，単純に残差を定義通りに計算する DC-GBiCGSTAB(s, L) 法と比べて，優位性は観とめられなかった．今後，前処理を適用した場合等における更なる有効性の調査が必要である.

表1 行列ベクトル積の数，CPU time，反復が終了したときの真の相対残差（行列 wang4，sme3Da の場合）

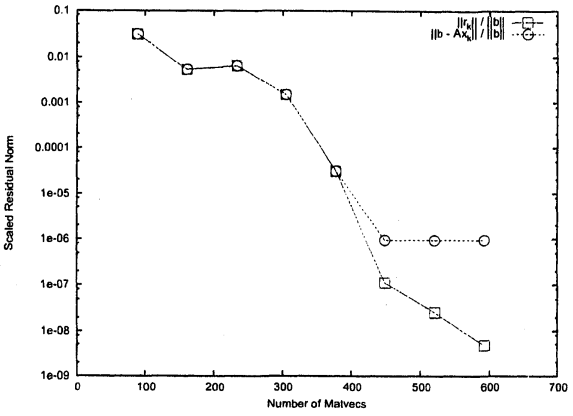
（表において，MV=行列ベクトル積の数，time(s)=CPU time(秒)，tnorm=反復が終了したときの真の相対残差，tnorm$\geq 10^{-8}$ の場合(すなわち偽収束の場合)tnorm をボールドで印刷している）

(a) wang4

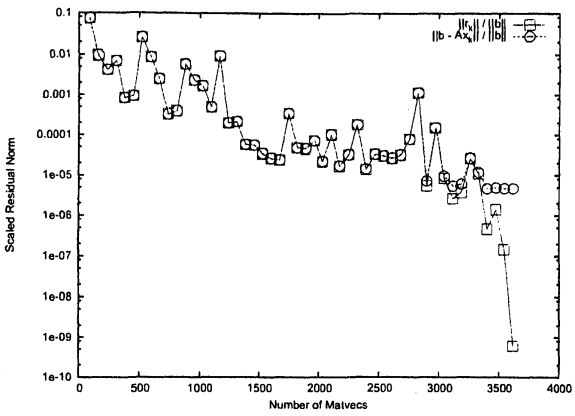
method	s	L	tnorm	MV	time(s)
GBiCGSTAB	2	1	1.52×10^{-9}	521	0.38
		2	2.94×10^{-9}	521	0.45
		4	5.67×10^{-9}	521	0.61
		8	7.60×10^{-9}	533	1.00
	4	1	3.27×10^{-9}	484	0.50
		2	4.57×10^{-9}	489	0.62
		4	6.63×10^{-9}	489	0.91
		8	$\mathbf{6.74 \times 10^{-8}}$	529	1.56
	8	1	7.39×10^{-9}	467	0.84
		2	5.88×10^{-9}	467	1.07
		4	3.06×10^{-9}	485	1.54
		8	$\mathbf{9.52 \times 10^{-7}}$	593	2.84
AC-GBiCGSTAB	2	1	9.16×10^{-9}	568	0.44
		2	4.93×10^{-9}	567	0.50
		4	4.54×10^{-9}	626	0.73
		8	1.37×10^{-9}	573	1.07
	4	1	4.71×10^{-9}	537	0.56
		2	1.42×10^{-9}	564	0.73
		4	9.19×10^{-9}	505	0.94
		8	2.26×10^{-9}	540	1.59
	8	1	4.58×10^{-9}	500	0.89
		2	6.95×10^{-9}	487	1.11
		4	3.12×10^{-9}	495	1.56
		8	4.93×10^{-9}	599	2.87
DC-GBiCGSTAB	2	1	2.27×10^{-9}	697	0.47
		2	6.60×10^{-9}	614	0.51
		4	8.01×10^{-10}	629	0.71
		8	2.89×10^{-9}	580	1.05
	4	1	8.02×10^{-9}	579	0.56
		2	1.24×10^{-9}	581	0.73
		4	8.75×10^{-9}	513	0.93
		8	2.28×10^{-9}	542	1.58
	8	1	7.45×10^{-9}	517	0.90
		2	7.78×10^{-9}	492	1.11
		4	3.10×10^{-9}	498	1.56
		8	6.20×10^{-9}	601	2.86

(b) sme3Da

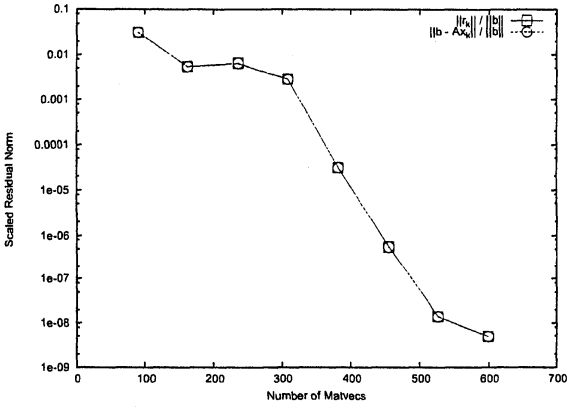
method	s	L	tnorm	MV	time(s)
GBiCGSTAB	2	1	7.32×10^{-9}	8087	30.68
		2	8.09×10^{-9}	5363	20.68
		4	9.62×10^{-9}	5105	20.33
		8	9.75×10^{-9}	6317	26.85
	4	1	7.46×10^{-9}	3254	12.76
		2	7.75×10^{-9}	3229	13.04
		4	9.52×10^{-10}	3489	14.79
		8	$\mathbf{1.23 \times 10^{-7}}$	4089	19.14
	8	1	3.70×10^{-9}	2771	11.68
		2	1.46×10^{-9}	2753	12.06
		4	7.98×10^{-9}	2825	13.43
		8	$\mathbf{4.69 \times 10^{-6}}$	3617	19.86
AC-GBiCGSTAB	2	1	5.72×10^{-9}	12240	46.79
		2	9.09×10^{-9}	5148	20.00
		4	5.57×10^{-9}	5118	20.34
		8	4.58×10^{-9}	6806	28.97
	4	1	2.42×10^{-9}	3401	13.45
		2	7.36×10^{-9}	3474	14.01
		4	7.37×10^{-9}	3572	15.16
		8	7.05×10^{-9}	4772	22.43
	8	1	2.31×10^{-9}	2879	12.17
		2	4.94×10^{-9}	2895	12.75
		4	1.20×10^{-9}	2943	14.03
		8	3.46×10^{-9}	3946	21.71
DC-GBiCGSTAB	2	1	9.18×10^{-9}	11229	41.91
		2	6.01×10^{-9}	6683	25.48
		4	6.63×10^{-9}	5543	21.89
		8	4.24×10^{-9}	8030	33.85
	4	1	1.38×10^{-9}	4911	19.08
		2	1.78×10^{-9}	3936	15.67
		4	5.23×10^{-9}	3705	15.64
		8	1.80×10^{-9}	4683	21.79
	8	1	4.42×10^{-9}	3037	12.65
		2	3.57×10^{-9}	3000	13.05
		4	3.29×10^{-9}	2977	14.08
		8	7.86×10^{-10}	3886	21.24



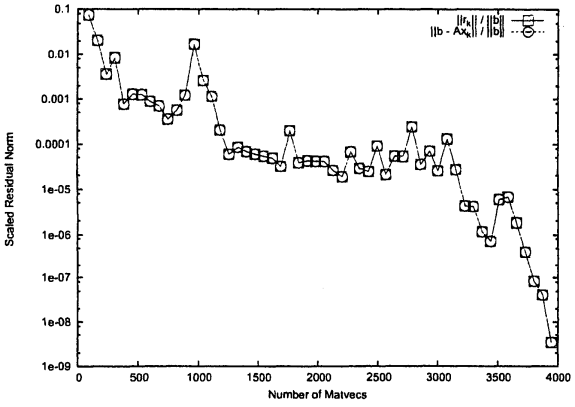
(a) GBiCGSTAB(8,8)



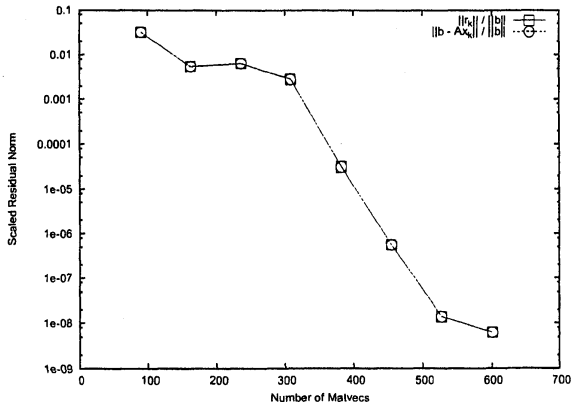
(a) GBiCGSTAB(8,8) 法



(b) AC-GBiCGSTAB(8,8)

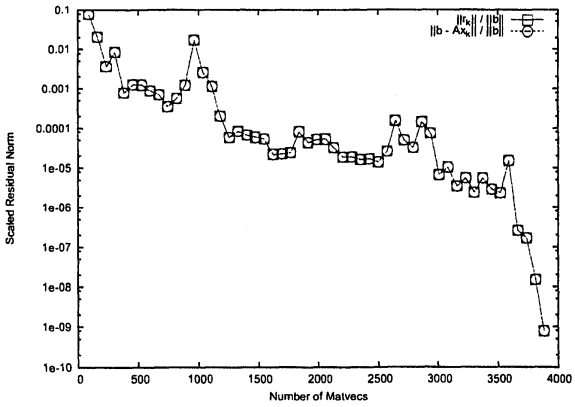


(b) AC-GBiCGSTAB(8,8) 法



(c) DC-GBiCGSTAB(8,8)

図 3 GBiCGSTAB(8,8), AC-GBiCGSTAB (8,8), DC-GBiCGSTAB(8,8) を行列 wang4 に適用したときの残差履歴



(c) DC-GBiCGSTAB(8,8) 法

図 4 GBiCGSTAB(8,8), AC-GBiCGSTAB (8,8), DC-GBiCGSTAB(8,8) を行列 sme3Da に適用したときの残差履歴

表 2 CPU time と真の解に収束した割合

(表において，PCC(percentage of correct convergences)
= $s, L = 1, 2, 4, 8$ として解法を適用したときの真の解に収束した場合の数の割合)

行列	次元	非零要素数	CPU time と真の解に収束した割合					
			GBiCGSTAB		AC-GBiCGSTAB		DC-GBiCGSTAB	
			time	PCC(%)	time	PCC(%)	time	PCC(%)
1138_bus	1138	2596	1.00	87.50	1.04	93.75	1.07	100.00
add20	2395	17319	1.00	93.75	1.08	100.00	1.08	100.00
add32	4960	23884	1.00	100.00	1.03	100.00	1.06	100.00
bcspwr06	1454	3377	1.00	100.00	1.06	100.00	1.06	100.00
bcsstk08	1074	7017	1.00	87.50	1.00	93.75	1.07	100.00
bcsstk14	1806	32630	1.00	93.75	1.04	93.75	1.09	100.00
cavity10	2597	76367	1.00	87.50	1.05	100.00	1.14	100.00
dwt_1005	1005	4813	1.00	93.75	1.14	100.00	1.03	100.00
epb3	84617	463625	1.00	93.75	1.05	100.00	1.09	100.00
eris1176	1176	9864	1.00	100.00	1.07	100.00	1.11	100.00
fidap022	839	22613	1.00	87.50	1.05	100.00	1.12	100.00
fidapm03	2532	50380	1.00	100.00	1.00	100.00	1.07	100.00
fs_541_4	541	4285	1.00	87.50	1.02	93.75	1.08	100.00
gr_30_30	900	4322	1.00	100.00	1.04	100.00	1.05	100.00
jagmesh2	1009	3937	1.00	100.00	1.08	100.00	1.09	100.00
lshp1270	1270	4969	1.00	100.00	1.05	100.00	1.09	100.00
memplus	17758	126150	1.00	81.25	1.07	93.75	1.08	100.00
orsirr_1	1030	6858	1.00	68.75	1.02	100.00	1.06	100.00
orsreg_1	2205	14133	1.00	75.00	1.04	93.75	1.17	100.00
poisson3Da	13514	352762	1.00	87.50	1.04	100.00	1.11	100.00
poisson3Db	85623	2374949	1.00	81.25	1.02	100.00	1.09	100.00
raefsky2	3242	294276	1.00	87.50	1.03	100.00	1.11	100.00
sherman1	1000	2375	1.00	93.75	1.02	100.00	1.04	100.00
sherman5	3312	20793	1.00	93.75	1.04	100.00	1.09	100.00
sme3Da	12504	874887	1.00	81.25	1.16	87.50	1.15	100.00
sme3Db	29067	2081063	1.00	81.25	1.00	100.00	1.07	100.00
torso3	259156	4429042	1.00	100.00	1.01	100.00	1.07	100.00
wang4	26068	177196	1.00	87.50	1.08	100.00	1.10	100.00
watt_1	1856	11360	1.00	87.50	1.02	87.50	1.06	100.00
watt_2	1856	11550	1.00	75.00	1.04	75.00	1.01	81.25
Average			1.00	89.79	1.05	97.08	1.08	99.38

参考文献

[1] T. Davis: University of Florida sparse matrix collection, http://www.cise.ufl.edu/ research/sparse/matrices/.

[2] MatrixMarket: http://math.nist.gov/MatrixMarket/.

[3] G. L. G. Sleijpen and D. R. Fokkema: BiCGSTAB(L) for linear equations involving unsymmetric matrices with complex spectrum, *Electronic Transactions on Numerical Analysis*, Vol. 1 (1993), pp. 11–32.

[4] 櫻井隆雄, 直野健, 恵木正史, 猪貝光祥, 木立啓之, 小路将徳: 高速性と信頼性を両立する AC-IDR(s) 法の提案と評価, 情報処理学会研究報告, 2008 (74), pp. 49–54.

[5] G. L. G. Sleijpen and M. B. van Gijzen: Exploiting BiCGstab(L) strategies to induce dimension reduction, *SIAM Journal on Scientific Computing*, Vol. 32 (2010). pp. 2687–2709.

[6] P. Sonneveld and M. B. van Gijzen: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations, *Reports of the Department of Applied Mathematical Analysis*, REPORT 07-07 (2007), Delft University of Technology.

[7] P. Sonneveld and M. B. van Gijzen: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, *SIAM Journal on Scientific Computing*, Vol. 31 (2008), pp. 1035–1062.

[8] M. Tanio and M. Sugihara: GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials, *Journal of Computational and Applied Mathematics*, Vol. 235 (2010), pp. 765–784.